# Lab 6 Report

ECE 332 Winter 2020

Jose Manuel Lopez Alcala & Benjamin Carlson

# 1 Introduction

In this lab, we learned how to characterize an induction motor using a second-order differential equation. For this, we also learned how to tune a PID loop in order to obtain the desired response from a step response. Using the PID loop enabled us to have a more precisely controlled motor and had less error than the previously designed systems that were based on discrete values.

# 2 Methods

## 2.1 Hardware

The hardware was connected identically as in the last lab. Figure 1 shows the board to which we connected the motor to.

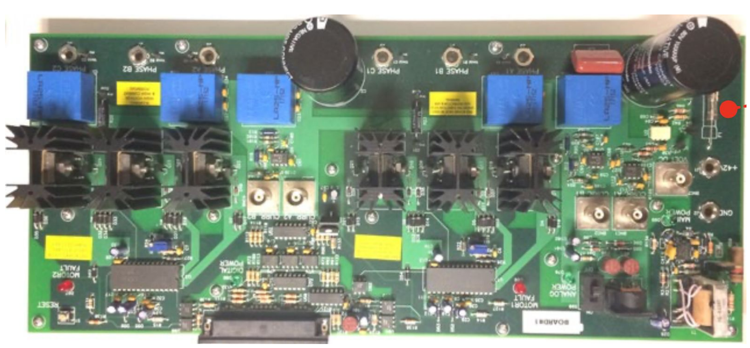

Figure 1. Controlling board for motor

The board was connected to the power supply that supplied the motor power. Additionally, it had another power supply that powered the logic on the board itself. This power supply is shown in Figure 2.

Figure 2. Power supply for logic on controlling board.

The three-phase ac induction motor was powered by 42 VDC from the power supply through an active bridge inverter in the board and output a three-phase sinusoidal PWM signal to create the rotating magnetic field required to control the speed of the motor.

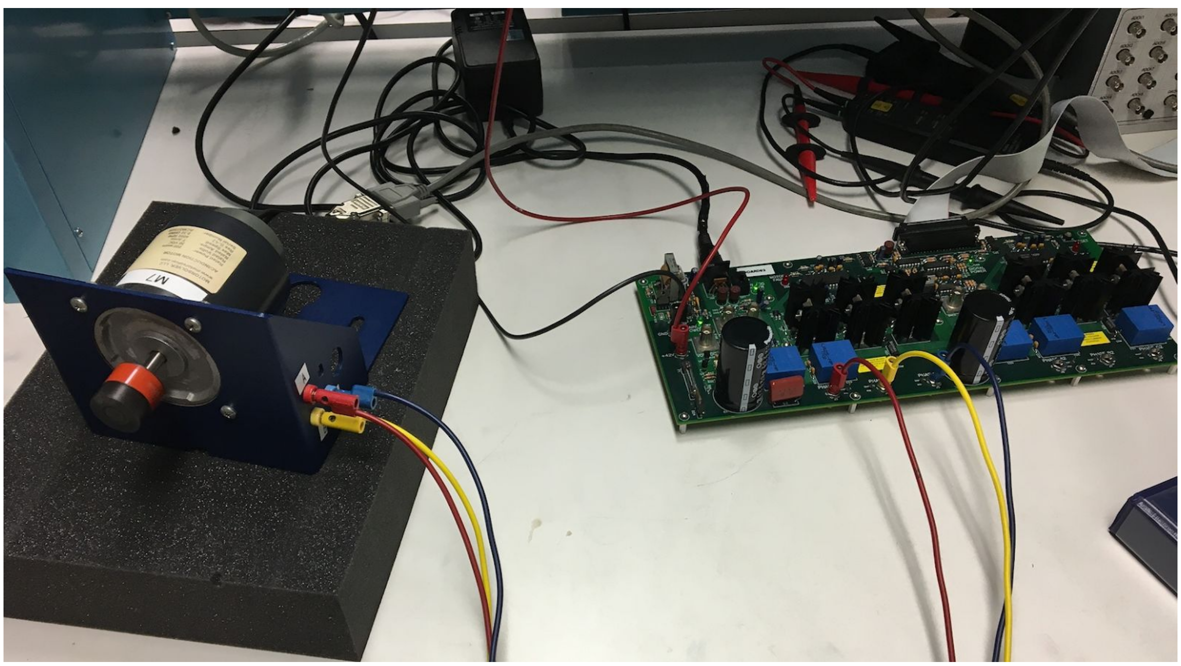

Figure 3. Experimental set-up.

## 2.3 Motor Characterization

The motor’s response to a step input must be sampled using an open-loop controller. This means that there is no compensation in the output of the motor response. This ensures that the output is the raw, uncompensated value. Figure 4 shows a theoretical output from an uncompensated output of the induction motor.

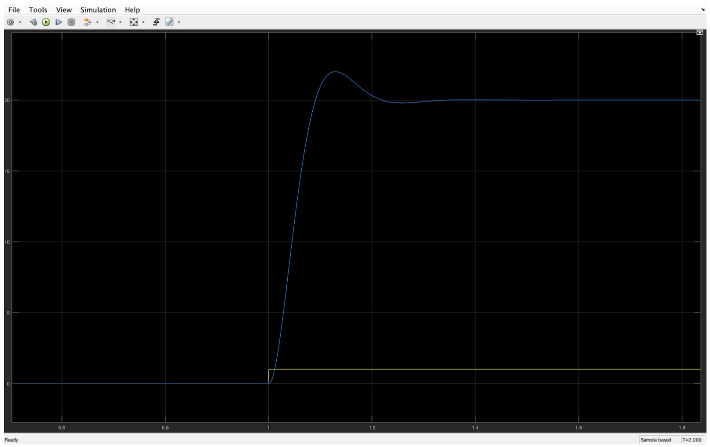


Figure 4. The theoretical output of induction motor response. Blue is output, yellow is the input step.

Using this principle, the characteristic equation for this motor would look like:

$$G(s) = \frac{K * {W_n}^2}{S^2 + 2\zeta S + {W_n}^2}$$

$$K = \frac{X_{ss}}{U_{ss}} = static\ gain; X_{ss} = Steady\ state\ value,\ U_{ss} = Step\ value$$

$$OS = \frac{X_p - X_{ss}}{X_{ss}} = overshoot; X_p = max\ value, X_{ss} = Steady\ state\ value$$

$$\zeta = \frac{-ln(OS)}{\sqrt{\pi^2 + [ln(OS)]^2}} = Damping\ constant$$

$$W_n = \frac{W_d}{\sqrt{1-\zeta^2}} = Natural\ Freq.; W_d = \frac{\pi}{T_p} = Damped\ Freq; T_p = Settling\ time$$

After pulling these values from the motor response, our equation with the correct coefficients is:

$$G(s) = \frac{318888}{S^2 + 113.5 + 10629}$$

This meant that we could now tune our PID block to enable us for better motor control.

## 2.4 PID Tuning

The PID tuning process involves having a function that is able to approximate the motor, this equation was found in section 2.3. Using this equation we can tune the PID loop. Figure 5 shows the setup for PID tuning.

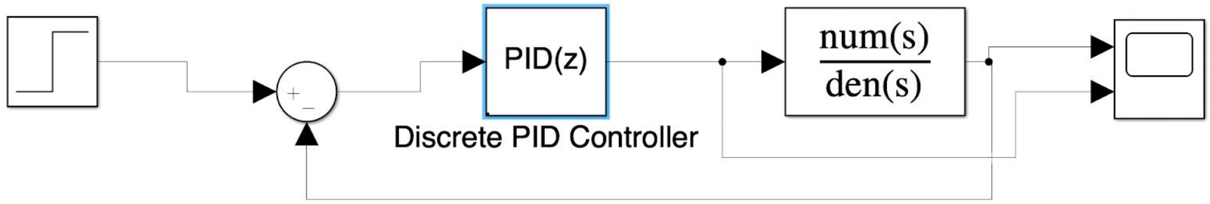


Figure 5. PID controller tuning setup.

After the tuning, we achieved the following coefficients as shown in Figure 6.

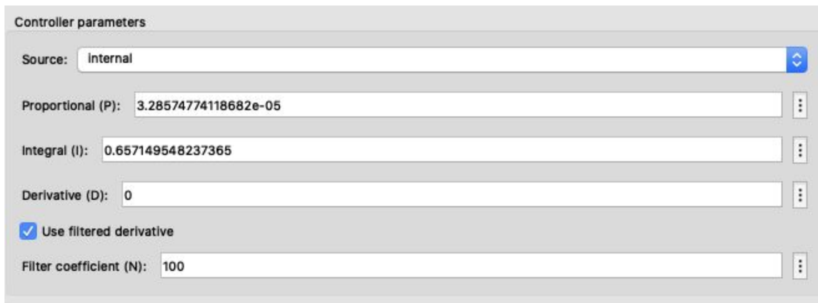


Figure 6. PID coefficients after tuning.

These coefficients were obtained by using the GUI sliders until the response satisfied the requirements that we were given.

## 2.5 Simulink Design

The final design replaces our discrete controller with the PID controller that we just tuned. We make use of our previous design from Lab 5 and swap our controller with the PID controller. Lab 5's design can be observed in Figure 7.

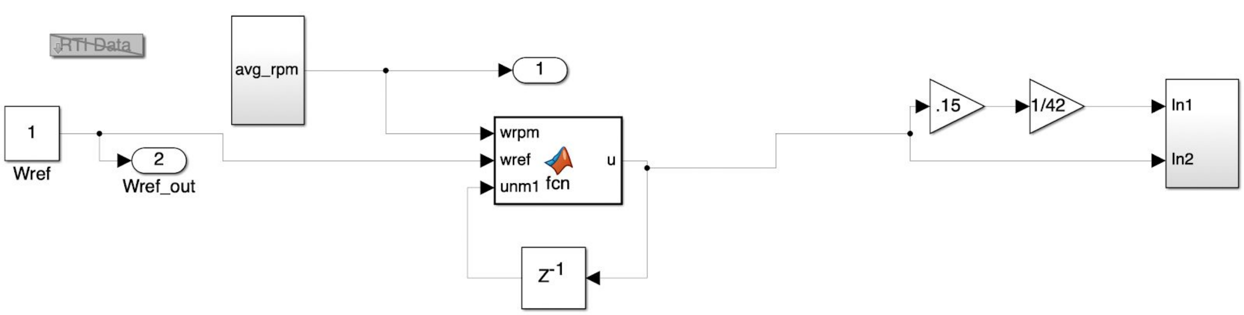


Figure 7. Lab 5 Final design that implemented a discrete controller.

Our final design for Lab 6 is shown in Figure 8. It can be seen that the controller has been swapped and it now makes use of the PID controller.

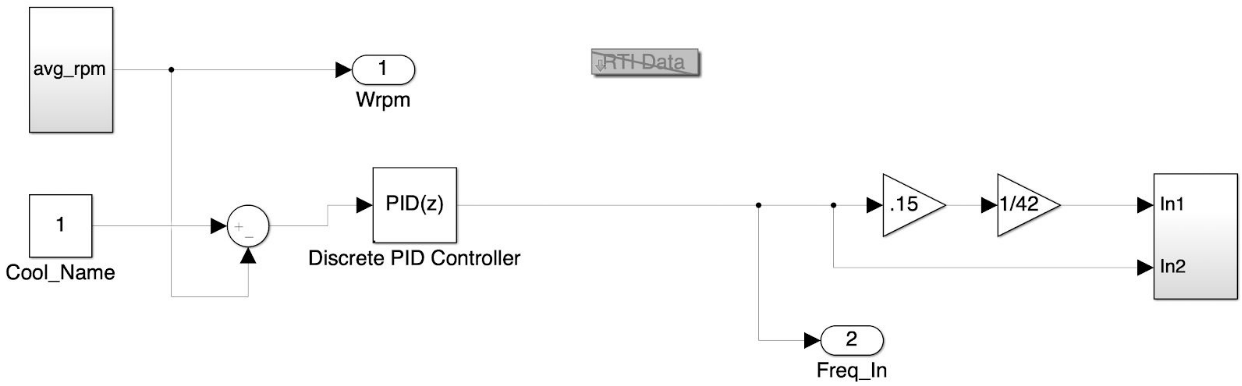


Figure 8. Final Simulink design after the integration of the tuned PID controller.

With the implementation of the PID controller, we expect to see a better response to output of the motor when the input is changed.

# 3 Results

The functionality of this lab's implementation resulted in a more responsive version of the motor in lab 5 by utilizing the PID loop. Using a stacked display on the graph for the measurements from the machine's encoder, the steady-state response time can be observed to be approximately 0.3 seconds, which is well within the 1 second response time required for this lab, and with a ripple of < 1% of the motor's reference speed. Contrasted with the results of lab 5 that operated in an open-loop discrete system, both of these parameters show significant improvement.

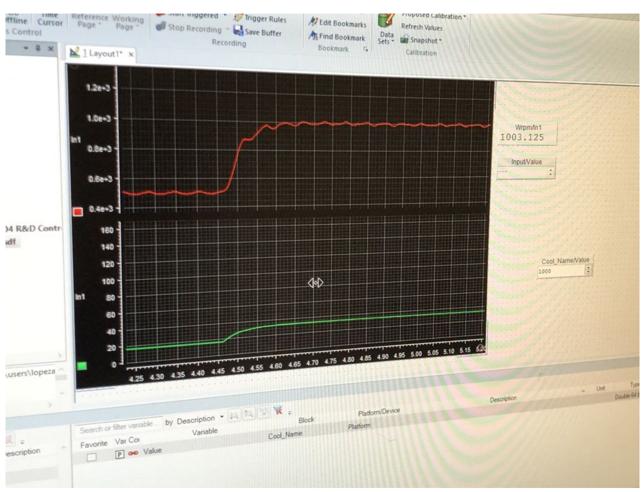

Figure 9. The graphed output of the implementation of Lab 6 final design.

# 5 Discussion

For this lab, we were able to successfully implement a second order transfer function for a 3-phase induction motor. The response of the machine was controlled by a PID block in Simulink with coefficients determined by the program to achieve the desired damping effect and transient delay, in a manner to be within the required values. The addition of the PID demonstrated a significant improvement in the motor’s response. The design has little room for improvement, with the transfer functions coefficients essentially controlling the motor’s response, any change in these values to improve one aspect of the response, would negatively affect another.